河南有线的创新应用

——日报数据的自动抓取与填报

摘　要：本文描述了使用 Python 抓取动态加载页面的报表数据、更新 Excel 模板数据的全过程，从而实现公司周期性报表数据的自动填报。重点介绍了两个问题的解决方案：一是怎样获取 JavaScript 动态加载页面的数据；二是怎样部分更新 Excel 模板的数据。

关键词：Python selenium Excel；自动填报
中图分类号：TP391　　文献标识码：A
文章编号：1671-0134（2019）12-060-03　　DOI：10.19483/j.cnki.11-4653/n.2019.12.016
本文著录格式：苏本国，张卫云 . 河南有线的创新应用——日报数据的自动抓取与填报 [J]. 中国传媒科技，2019（12）：60-62.

文 / 苏本国　张卫云

1. 背景

麦肯锡称：数据，已经渗透到当今每一个行业和业务职能领域，成为重要的生产因素。在这个时代，公司的决策者、经营者都需要通过数据观察企业运作状态以及规律，没有数据，我们举步维艰。

系统报表能为我们提供各种基础数据，但数据维度和格式固化，不能灵活满足我们的临时需求。所以，手工填报定制的、多变的经营日报（周报 \ 月报）是所有经营单位都必须持续开展的日常工作。

Excel 灵活而强大，能处理工作中大部分的数据。使用 Excel 可以方便地制作包含原数据、计算过程和最终展现结果的日报（周报 \ 月报）模板。

河南有线信息支撑部在日常工作中，常根据公司领导要求，临时制作各种 Excel 模板，并根据当时需求，有选择地将各系统平台上的报表数据手工填入临时模板。该工作难度低，重复性强，尤其月初，需要填报的模板在 30 份左右，每份需要打开的报表页面基本都在 10 个以上，仅月初就需要 6 人一天的工作量。

为此，我们尝试寻找一种自动获取数据并填报的方法，最终找到目前排行第一的开源开发工具 Python，经过一段时间的学习和研究，我们利用该工具编写了网页爬取、Excel 数据填写的可执行程序，成功实现了数据的自动填报工作目标。

2. 思路和过程概述

图 1

使用 python 编写网页爬虫的步骤，可分为：发送请求、获取数据、解析数据和存储数据四步。但在实际操作中，因爬取方法不同，我们直接获取到了数据的列表，所以省去了数据解析环节，只保留了三步：发送请求、获取数据和存储数据。每个过程处理细节详情如图 1 所示：

下面按图示步骤，分步说明在应用程序的编写过程中，每一步遇到的问题及相应的解决方案。

3. 编制过程详解

3. 1 发送请求

3.1.1 请求页面的三种方法选择

爬取数据，第一步操作就是模拟浏览器向网页所在的服务器发出请求。我们需要抓取的页面是公司内部 CRM 客户关系管理系统的报表数据，无需身份验证，但请求参数较多，报表数据由 JavaScript 动态加载。

3.1.1.1 基础的用法，使用 urllib 的 request 模块

该模块中的 urlopen（ ）方法，可以实现简单的请求发送操作，并得到响应。但该方法在构造带参数的请求时较为复杂。

3.1.1.2 高级用法，使用 requests 库

requests 库中的方法可轻松实现带参数、cookies、登录验证和代理设置等网页请求，但得到的结果和在浏览器中看到的不一样：在浏览器中可以看到的显示数据，但 requests 得到的结果中并没有。这是因为 requests 获取的都是原始 HTML 文档，而浏览器中的数据则是经过 JavaScript 处理数据后生成的结果。

3.1.1.3 模拟浏览器法，使用 selenium 库

为了解决获取 JavaScript 生成的动态页面数据问题，我们查阅相关资料后最终选择使用模拟浏览器库——Selenium 处理。

Selenium 是一个自动测试工具，利用它可驱动浏览器执行特定的动作，如点击、下拉等操作，同时还可以获取浏览器当前呈现的页面源代码，做到可见即可爬。注：在使用该方法前，除安装 Selenium 库外，还需要正确安装好使用的浏览器，如 Chrome，并配置好 ChromeDriver。

3.1.2 页面分析与构造请求

分析请求参数，打开报表页面，按 F12 打开“开发者工具”，从 Query String Parameters 发现 URL 中所带参数较多，且含有中文参数（地市信息）。针对这样的复杂参数信息，我们采用了“基础地址 + 参数信息”的方法重新构造 URL，然后再使用 Selenium 库发送请求。在此过程中，遇到了不少细节问题，详情及解决方法如表 1 所示：

表 1

问题描述	解决方法
中文参数	使用 quote（ ）和 unquote（ ）转码
将参数与基础地址拼接形成可用的 URL	使用“基础地址 +urlencode（参数）”
参数中有特殊字符“/”	使用 safe 如：urlencode（params，’utf-8’，safe=’/’）

3.1.3 发送请求

根据“分析请求参数”时所获信息，使用 selenium 库模拟谷歌浏览器向服务器发出请求。获取各地市现金流的脚本编写如下：

```
def get_cash（p_r_name，p_r_id，std，edd，c_id，p_type，p_id，p_name，p_r_type）：
    browser=webdriver.Chrome（ ）# 初始化一个浏览器
    base_url = ‘http：//…/bossreport25/frameset?’# 基础地址
    parames = {‘参数 1’：p_r_name# 传递前台输入的参数
    ……
    ‘参数 n’：p_r_type}
    url = base_url + urlencode(params，’utf-8’，safe=’/’)# 重构带参数的地址
    browser.get（url）# 发送请求
```

3. 2 获取数据

3.2.1 页面分析、获取数据

打开报表页面，按 F12 打开“开发者工具”，选中要提取的元素，右键选择“审查元素”，可找到该元素所在的节点位置。由于该元素没有较明确的节点 ID，且有较多同类节点，因此采用逐级上查，找到离其最近的有明确节点 ID 的节点“__bookmark2__”，以便 CSS 选择器定位待查数据。获取数据脚本如下：

```
bookmark =browser.find_element_by_id(“__bookmark_2”)# 找到指定元素
cash_list = bookmark.find_elements_by_css_selector(“tr”)# 找到该元素包含的数据元素
```

3.2.2 关于延时等待

在调测过程中，发现报表页面自动打开后很快关闭，并没有获得目标数据。资料显示 selenium 的 get（ ）方法会在网页的框架加载结束后结束执行，此时，服务器给浏览器的响应中可能也没有目标数据。因此，这里需要增加延时等待。延时等待分显示和隐式，在本应用中，我们采用了显示等待的方法，在控制语句中增加了 WebDriverWait() 函数。即：在规定时间内加载指定节点，如果加载完成，则正常返回查找的节点，否则，抛出超时异常。控制脚本如下：

```
wait =WebDriverWait（browser，20）# 等待指定时间
```

wait.until（EC.presence_of_all_elements_located（By.ID，‘__bookmark_2’）））# 显示等待，直接到指定元素载入

3.3 存储数据

3.3.1Python 往 Excel 中写数据的 5 种方法

Python 拥有一个强大的标准库，同时，Python 社区提供了大量的第三方模块。完成一项任务可有多种方法，只有选择合适的方法才能达到自己的目标。将所获数据写入 Excel 时，我们尝试了多种方法，但都无法实现“无损模板地更新”的目标。网上有文章整理了 Python 写入 Excel 的 4 种方法及其优缺点，增加我们自己的一种方法，归纳如下：

表 2

方法	to_excel	XlsWriter	xlrd&xlwt	OpenPyXL	Microsoft Excel API
简介	将 dataFrame 中数据导出至 Excel 表	可创建 Excel 2007 或更高版本的 XLSX 文件	含 xlrd、xlwt 模块，分别提供读、写功能	可以读写 XLSX 和 XLS 文件	直接通过 COM 组件与 Microsoft Excel 进程通信，通用其各种功能实现对文件的操作
读取	√	×	√	√	√
写入	√	√	√	√	√
修改	×	×	×	√	√
.xls	√	×	√	×	√
.xlsx	√	√	慎用	√	√
系统限制	无	无	无	无	Windows+Excel

其中，使用 OpenPyXL 修改模板时，只可追加 sheet 页，但不能更新单元格，会影响表中原有公式；但使用 Microsoft Excel 则可修改部分单元格数据，且不会影响原公式。因为本应用中既需要读，又需要更新 Excel 文档中的部分数据，且不能修改原文档中的公式，所以，在此只能使用 Microsoft Excel API，即引用 win32com 组件。

3.3.2 使用 win32com 组件，修改 Excel 表中部分数据

写入 Excel 文档的全过程：调用 win32com 组件，启动独立的 Excel 进程，并打开 Excel 模板文件，使用 sheet.Cells（i，j）.Value 实现给“第 i 行第 j 列”单元格赋值。相关脚本如下：

excel = win32com.client.DispatchEx（'Excel.Application'）# 启动独立的 Excel 进程

cash_book= excel.Workbooks.Open（‘F：/ 模 板 .xlsx’，ReadOnly=False）# 打开模板

sht1=cash_book.Worksheets（‘sheet1’# 打开待更新的 sheet 页

for i in range（0，18）# 18 分公司，需要读 18 行数据

for j in range（0，6）：# 每个分公司需要 6 项数据

sht.Cells（i+5，j+4）.Value=cash_list[8*i+j] # 需要从当前 sheet 的第 5 行第 4 列更新数据，数据来源为以上获取的数据列表

j += j

i += i

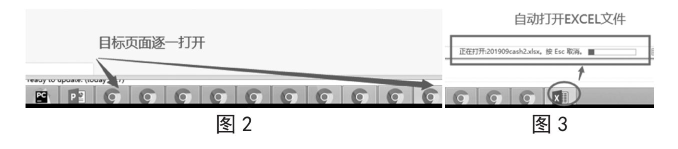


图 2　图 3

4. 成果展现

经过以上三步操作，一个完整的数据抓取和填报程序就完成了，加上友好的参数录入及进度提示住处，再使用 pyinstaller 将程序编译成可执行文件。将可执行文件和模板一起移置到应用环境中，按周期执行该文件，输入统计需要的参数，即可以直观地看到页面打开过程和 Excel 数据刷新过程。图 2 为目标页面逐一打开的过程，图 3 为 Excel 文档自动打开的过程，此后，随着 Excel 数据的刷新，文档中原有计算公式会自动计算，待数据写入完毕，目标数据即为可发布数据。

结语

使用 python 的扩展库和模块实现获取和使用数据较为简单，本应用使用的扩展库和模块有：selenium、urllib、win32com 和 datetime 等。方便快捷地实现了目标功能，解决常用报表的自动填报问题，在节约人力成本的同时，提高了工作效率和数据准确度。媒

参考文献

[1] 张俊红 . 对比 EXCEL，轻松学习 Python 数据分析 [M]. 北京：电子工业出版社，2019.

[2] 崔庆才 .Python3 网络爬虫开发实战 [M]. 北京：人民邮电出版社，2018.

[3] CSDN 博客：https：//blog.csdn.net/SvJr6gGCzUJ96OyUo/article/details/78967060.

（作者单位：河南有线电视网络集团有限公司信息化支撑部）